The DeepScoresV2 Dataset and Benchmark for Music Object Detection

Conference Paper · January 2021

DOI: 10.1109/ICPR48806.2021.9412290

Lukas Tuggener
Zurich University of Applied Sciences
15 PUBLICATIONS **91** CITATIONS

Alexander Pacha
TU Wien
16 PUBLICATIONS **164** CITATIONS

Thilo Stadelmann
Zurich University of Applied Sciences
68 PUBLICATIONS **419** CITATIONS

The DeepScoresV2 Dataset and Benchmark for Music Object Detection

Lukas Tuggener*
ZHAW Datalab & USI
Winterthur & Lugano, Switzerland
tugg@zhaw.ch

Yvan Putra Satyawan*
ZHAW Datalab
Winterthur, Switzerland
https://orcid.org/0000-0002-6375-8308

Alexander Pacha
TU Wien
Vienna, Austria
alexander.pacha@tuwien.ac.at

Jürgen Schmidhuber
The Swiss AI Lab IDSIA (USI & SUPSI)
Manno-Lugano, Switzerland
juergen@idsia.ch

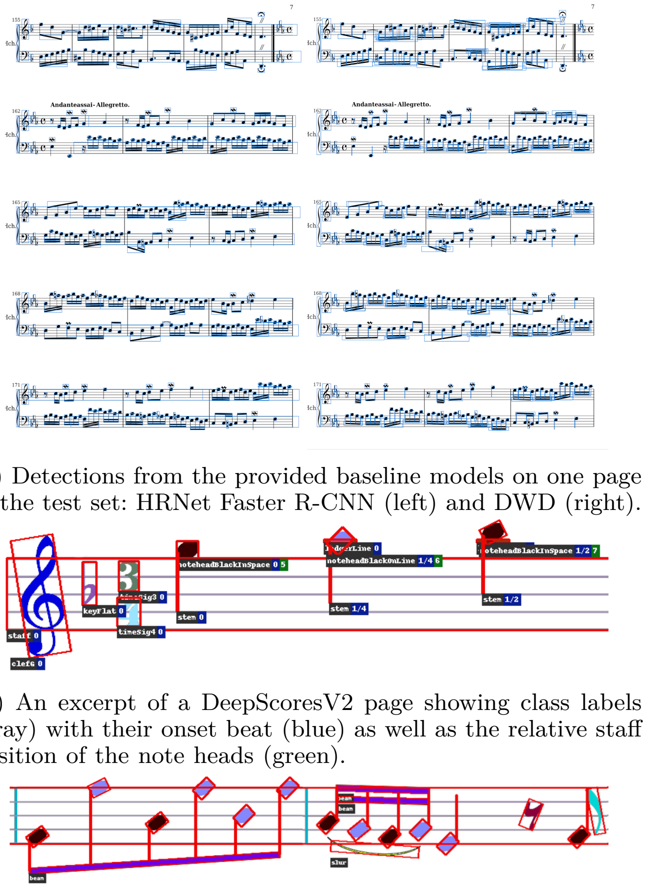
Thilo Stadelmann
ZHAW School of Engineering
Winterthur, Switzerland
stdm@zhaw.ch

Abstract—In this paper, we present DeepScoresV2, an extended version of the DeepScores dataset for optical music recognition (OMR). We improve upon the original DeepScores dataset by providing much more detailed annotations, namely (a) annotations for 135 classes including fundamental symbols of non-fixed size and shape, increasing the number of annotated symbols by 23%; (b) oriented bounding boxes; (c) higher-level rhythm and pitch information (onset beat for all symbols and line position for noteheads); and (d) a compatibility mode for easy use in conjunction with the MUSCIMA++ dataset for OMR on handwritten documents. These additions open up the potential for future advancement in OMR research. Additionally, we release two state-of-the-art baselines for DeepScoresV2 based on Faster R-CNN and the Deep Watershed Detector. An analysis of the baselines shows that regular orthogonal bounding boxes are unsuitable for objects which are long, small, and potentially rotated, such as ties and beams, which demonstrates the need for detection algorithms that naturally incorporate object angles. The dataset, code and pre-trained models, as well as user instructions, are publicly available at https://zenodo.org/record/4012193.

Index Terms—Optical music recognition, deep neural nets, music object detection

I. Introduction

Optical music recognition (OMR) is the research field concerned with computationally reading musical notation in documents [1]. It is a challenging sub-field of computer vision and document recognition, with the goal to convert scanned music sheets into a machine-readable format for further processing. A crucial sub-task of OMR is the localization and classification of individual symbols of music notation, also referred to as music object detection. A core difference between object detection in real-world photos and music object detection is the number of objects that usually appear in a single image. While there are tens of objects in natural images, it is not uncommon to have hundreds or even thousands of objects of interest in a single music score image. Additionally, music symbols often rely heavily on the context to be classified correctly.



(a) Detections from the provided baseline models on one page of the test set: HRNet Faster R-CNN (left) and DWD (right).

(b) An excerpt of a DeepScoresV2 page showing class labels (gray) with their onset beat (blue) as well as the relative staff position of the note heads (green).

(c) An excerpt of a DeepScoresV2 page showcasing some of the newly annotated variably sized symbols (beams, slur) together with their oriented bounding boxes.

Fig. 1: Overview of novelties in DeepScoresV2: ground-truth and (a) predictions for full hi-res pages from two baselines, (b) rhythm and pitch annotations, and (c) new variably sized symbols with oriented bounding boxes. Not shown: compatibility mode with other OMR datasets.

*) The first two authors contributed equally to this work.

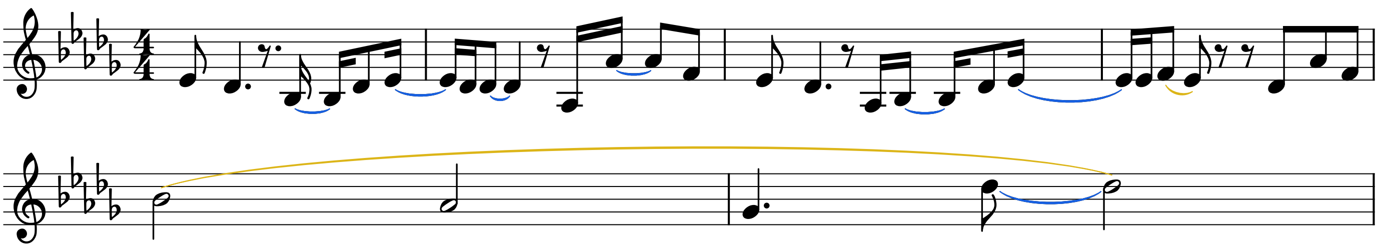


Fig. 2: Slurs (yellow) and ties (blue) can vary significantly in size, ranging from relatively short instances (top) to almost as wide as the entire staff (bottom). Depicted music are excerpts of “You make it real” by James Morrison.

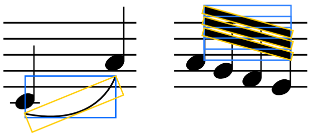


Fig. 3: Two examples of slanted symbols with their corresponding orthogonal bounding boxes (blue) and oriented bounding boxes (yellow). Orthogonal bounding boxes contain a significant amount of background pixels (left) and can have ample overlap with other bounding boxes (right). Oriented bounding boxes reduce these issues.

Previously, the Deep Watershed Detector was proposed to specifically address these issues [2]–[4]. Here, as with many other computer vision tasks, deep learning [5] has brought about significant advances to OMR, especially to the initial stages that visually process the image [6]–[8]. The need for sufficiently large, annotated datasets was first addressed with the release of the DeepScores dataset [9] (see Sec. II), which includes annotations for the subset of fixed-shape musical symbols, but does not ship with established benchmark results and pre-trained models for easy comparison. It also does not interface easily with other existing OMR datasets.

In this paper, we present DeepScoresV2, an extended and improved version of the original DeepScores dataset that specifically addresses these issues and makes the following contributions: we (a) add 20 formerly absent classes including symbols without fixed size or shape but are nonetheless fundamental to music notation, thereby increasing the list of musical symbols that can be detected by 23%; (b) add ground truth for oriented bounding boxes, thus enabling research into detectors with potentially much higher precision; (c) add ground truth for further higher-level musical semantics, therefore making the dataset valuable for tasks beyond pure music object detection downstream in OMR; (d) add a compatibility mode for DeepScoresV2 and MUSCIMA++ such that the two datasets can be easily used in conjunction; and (e) provide pre-trained state-of-the-art detectors and benchmarking results for comparisons.

The original DeepScores dataset was designed with only fixed-shape symbols in mind. In DeepScoresV2, the available classes additionally include variably-shaped symbols, such as beams and slurs, which can be as small as spanning two closely neighboring objects to being as wide as the entire page, as shown in Figure 2. This makes DeepScoresV2 not only more complete, but also makes achieving a high precision much more challenging.

Musical notation contains symbols that tend to have a very high width-to-height ratio and are non-orthogonal to the image axes. This leads to orthogonal bounding boxes that contain a large number of background pixels (see Figure 3, left) and, even more problematic, to bounding boxes that overlap largely with other bounding boxes (see Figure 3, right).

To address this issue, DeepScoresV2 contains both orthogonal bounding boxes as well as oriented bounding boxes that cover the minimum rectangular area around each object, as illustrated in Figure 3 (yellow). This ensures that bounding boxes represent their corresponding objects more accurately and reduce the amount of overlap with other objects. A quantitative analysis shows that DeepScoresV2 indeed represents symbols more accurately through its oriented bounding boxes that cover on average 13.34% less background.

Due to the high complexity of musical notation, OMR datasets generally have different, non-compatible annotations. This makes working on different datasets very complex and laborious. To enable easy interoperability with MUSCIMA++, we ship DeepScoresV2 with a compatibility mode that allows for out-of-the-box mixing of the two datasets. This is desirable for increased diversity (see Sec. II: one dataset is hand-written, one typeset).

Finally, we present baseline object detection algorithms on DeepScoresV2 which show that while some symbols in the extended symbol set can be detected reasonably well with state-of-the-art models, future work will be needed to achieve good detection on all, especially the new, symbols. Nevertheless, the baseline models and the experimental setup used are ready to use for any future study for comparisons.

II. Related Work

The recognition of music scores can be divided into different sub-problems such as detecting staff lines, detecting objects, and reconstructing semantics. In the past, several OMR datasets have been published that address one or more of these sub-problems. The “OMR Datasets Project” lists the most prominent ones and is updated regularly [10]. The most comprehensive datasets are:

- DeepScores [9] is a huge synthesized dataset of typeset music for large-scale music object detection and image segmentation. It consists of around 300, 000 pages of music scores with the corresponding annotations for detection and image segmentation. The dataset was generated by rendering existing MusicXML files with Lilypond into annotated SVG images. This process allows the generation of bounding box annotations as well as semantic segmentation masks. DeepScores is specifically designed for developing and evaluating systems that perform music object detection.
- MUSCIMA++ [11] is a small dataset of 140 images containing handwritten music notation. Detailed annotations are encoded in a Music Notation Graph [12], [13] including bounding boxes, class labels, and image masks for all primitives. Additionally, the graph models the syntactic relationships between the primitives as directed edges. It is based upon the CVC-MUSCIMA [14] dataset, which contains 20 carefully selected musical pieces, copied by 50 different musicians, totaling to 1, 000 images.
- HOMUS [15] is a large dataset for music symbol classification. It records 15, 000 samples of isolated music symbols with the individual strokes that were used to draw each symbol, allowing to perform online symbol classification.
- PrIMuS [16] is a large synthesized dataset of more than 87, 000 single-stave, monophonic musical snippets, rendered from their underlying MEI sources. It was extended into the Camera-PrIMuS [16] dataset by distorting the images to simulate an imperfect image capturing process.
- MSMD [17] is a medium, synthetic dataset of nearly 500 pieces of classical music with aligned note-head annotations between the score image and the corresponding MIDI file. It can be used for cross-modal retrieval scenarios such as score-following.
- DOTA [18] is a large dataset with over 2, 800 images for detecting objects in aerial imagery. While not a dataset for OMR, it shares many characteristics in the sense that it contains high-resolution images that depict hundreds of tiny objects per single image and in that it makes use of oriented bounding box annotations.

Most other datasets are either too small to draw statistically meaningful conclusions, lack proper annotations, or contain musical material that is protected by copyright laws, prohibiting their publication. These long-standing hindrances to progress in the field have largely been addressed in recent years and large, freely available datasets are becoming the norm. DeepScores and DeepScoresV2 are no exception, and, to the best of our knowledge, are the largest available OMR datasets for typeset music (see also Table I for a quantitative comparison with other general object detection datasets).

Dataset	Classes	Images	Object Inst.	Avg. Inst. per Image
PASCAL VOC [19]	20	21, 503	62, 199	2.89
COCO 2014 [20]	80	123, 287	886, 266	7.19
ImageNet [21]	200	349, 379	478, 806	1.37
DOTA [22]	15	2, 806	188, 282	67.10
MUSCIMA++V2 [23]	163	140	102, 914	735
DeepScoresV2	136	255, 385	151M	592
↪ dense	136	1, 714	1.1M	660

TABLE I: Comparison between DeepScoresV2 and other object detection datasets. Note the huge increase in both annotations and average annotations per image.

III. The DeepScoresV2 Dataset

Object detectors that are pre-trained on natural images result in poor performance when used for music object detection [9]. This is due to the following challenges:

- Large scale (size) variations both between different classes of symbols and between different instances of a single class. For example, some symbols, like slurs, are dynamically sized according to their contextual meanings while maintaining the same class.
- A large number of symbols on each page of sheet music. Typically, most object detection dataset contain in the range of tens to hundreds of instances per image. In contrast, most sheet music pages contain between a few hundred to thousands of individual objects per page.
- Many very thin symbols, which are not aligned with the axes of the image. This causes orthogonal bounding boxes to be an imprecise representation of musical symbols, containing more background than foreground pixels in each bounding box.

To address these challenges, we present DeepScoresV2, a large-scale, high-quality, fully annotated optical musical recognition dataset. DeepScoresV2 consists of 255, 386 pages of digitally engraved sheet music, rendered at 400 dots per inch (DPI) with tens of millions of symbols. We also provide a dense version of this dataset consisting of 1, 714 of the most diverse and challenging images split into 1, 362 training images and 352 test images. Annotations are also provided with the option of using multiple category names to allow for compatibility with the MUSCIMA++ dataset [11]. This is done so that cross-modal validation of techniques could be performed on both printed and handwritten music scores. Finally, we have excluded those pages from DeepScoresV2 that have malformed annotations in DeepScores to reduce the chances that incorrectly labeled annotations would appear in DeepScoresV2. Images are provided as PNG files along with segmentations in indexed PNG files, instance segmentation in PNG files, and annotations in JSON files.

Class Name	Class average background pixel ratio (%)		
	Orthogonal BBox	Oriented BBox	Improvement
slur	92.69	86.30	6.89
tie	84.48	78.83	6.69
clef8	77.73	54.03	30.49
beam	35.40	11.73	66.86
noteheadBlack	25.98	17.00	34.57
rest16th	66.05	54.98	16.76
Overall	55.83	49.26	13.34

TABLE II: Average background area reduction for selected classes and average overall reduction. “Background pixel ratio” shows what percentage of pixels within a bounding box is part of the background rather than the foreground.

A. Oriented Bounding Boxes

One of the main new features of DeepScoresV2 are the oriented bounding boxes. The area outlined by an orthogonal bounding box often contains a significant amount of background pixels, especially when the respective symbol is thin and slanted like a beam or slur. To address these shortcomings, we have added oriented bounding boxes to DeepScoresV2 labelled as 8-tuples $[x_0, y_0, x_1, y_1, x_2, y_2, x_3, y_3]$. These bounding boxes are always rectangular, but generally at an angle relative to the image axes, and calculated from the minimum area rectangle around each object instance as follows: using the PNG pixel array and the original DeepScores annotations that contain orthogonal bounding box information for every symbol, we calculate the oriented bounding box by treating each pixel of a symbol within the orthogonal bounding box as a point in a 2D space and effectively turn the problem into finding the minimum area rectangle around this set of points. This is finally calculated using the minimum area rectangle function provided in the Shapely package[1].

Qualitatively, these oriented bounding boxes are better representations of their objects as they more clearly depict the shape of the object, as seen in Figure 3. Quantitatively, we can reduce the number of background pixels contained within a bounding box by an average of 13.34%. A detailed analysis of some prominent classes is depicted in Table II.

For easy use of this bounding box scheme, we are making available the OBBAnns toolkit[2] as a framework-agnostic tool to work with the DeepScoresV2 dataset. It provides abstractions to load annotations, get image-annotation pairs by index or image ID, visualize the dataset, and calculate validation metrics, with the most computationally intensive operations implemented in C++. The toolkit can also be used to work with any dataset containing both oriented bounding boxes as well as ground-truth segmentation. The data schema and further instructions on how to use the toolkit can be found in the respective repository.

Symbol	Change with respect to DeepScores
beam	Added
clef	Changed all symbols to use clefX naming scheme and removed ”changed” suffix
staff	Added
hairpin	Added dynamicDiminuendoHairpin and dynamicCrescendoHairpin
dynamics	Changed to individual symbols, e.g. dynamicS, dynamicF, dynamicZ
ledgerLine	Added
noteheads	Added InSpace and OnLine suffixes
ottavaBracket	Added
restHNr	Added
restLonga	Removed
restMaxima	Removed
slur	Added
stem	Added
tie	Added
timeSig	Changed to individual numerals, e.g. timeSig0, timeSig1
tremolo	Added tremolo0 - tremolo5
tuplet	Added tuplet1 - tuplet9
tupletBracket	Added

TABLE III: Summary of changes to symbol classes in DeepScoresV2. Most notably is the addition of hairpins, beams, slurs, and ties. Additionally, some names have been changed to become more consistent, and certain compound symbols have been split into their component symbols for added robustness. Classes that do not occur in the dataset have been removed.

B. Extended Symbol Set

DeepScoresV2 introduces an extended symbol set encompassing variably sized symbols, including some changes for added musical context and having a few name changes to become more self-consistent (see Table III for a detailed overview). By incorporating variably sized symbols, a richer musical representation can be extracted as opposed to using the original set of classes, which contains only fixed-sized symbols. Symbols such as slurs and ties, which may span from two neighboring notes and up an entire line of music, as seen in Figure 2, are particularly difficult for machine learning algorithms to understand as a single class due to their scale variability. Newly introduced symbols from this category are beams, dynamicDiminuendoHairpins, dynamicCrescendoHairpins, slurs, stems, and ties.

New contextual symbols are also introduced as part of the extended symbol set, namely the stem, tuplet, tupletBracket, ottavaBrackets, ledgerLines, and tremolo classes. Finally, some symbol names are changed to be more consistent: for example, compound dynamic symbols and time signatures have all been reduced to their components and clef names have been rectified to be in line with dynamics, flags, and rests.

[1] https://github.com/Toblerity/Shapely
[2] https://github.com/yvan674/obb_anns

C. Additional Features

Apart from the aforementioned major contributions, there are numerous smaller additions included in DeepScoresV2:

1) Cross-dataset compatibility: Compatibility between OMR datasets has long been neglected, which has made it very difficult to compare different approaches and reuse existing work. To alleviate this problem, we define a compatibility mode that allows us to jointly use the MUSCIMA++ and DeepScoresV2 datasets, e.g., for model training or evaluation. The MUSCIMA++ dataset was chosen because it is, to the best of our knowledge, the only large OMR dataset which contains annotations on a similar level. Furthermore, the underlying material—handwritten music scores in modern notation—is a great complement for the DeepScoresV2 dataset. Compatibility is enforced by (a) confining the symbol sets to the subset of classes that appear in both datasets; (b) choosing a decomposition of musical symbols into detectable objects that both datasets can provide; and (c) aligning the class names wherever possible by following the SMuFL [24] conventions.

2) Staff information: DeepScoresV2 introduces additional information regarding the position of the notes with respect to the staff to facilitate pitch recognition. All notehead classes are split into -InSpace and -OnLine subclasses, making subsequent position-based pitch detection more robust against minor perturbations. For direct staff detection, every note head in DeepScoresV2 has its relative staff position stored in its annotation as an additional field.

3) Onset information: To enable research of OMR models with a deeper musical understanding, DeepScoresV2 also contains annotation for temporal onset for every symbol (on which beat a given symbol starts). This allows for the training of models capable of much higher level reconstruction of the music than just localization and classification of individual objects.

4) Instance segmentation annotations: While DeepScores already contains pixel-wise semantic segmentation ground truth, DeepScoresV2 ships with additional instance segmentation masks. We provide instance segmentation in separate PNG files containing instance information in the RGB-channels, starting from 1 and reset with every page. The instance number is encoded in the hexadecimal color value used (e.g. instance 1 has a color value of #000001). An example of an instance segmentation mask is shown in Figure 4.

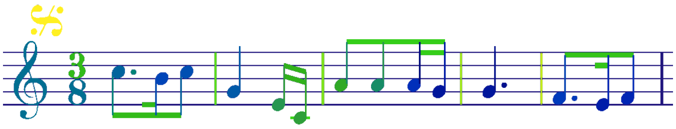


Fig. 4: An example of the instance segmentation ground truth. Every symbol occurrence has its own color due to the encoding of instance information as color values. Color differences have been exaggerated for better readability.

IV. Baseline Results on DeepScoresV2

To highlight some of the peculiarities of the dataset as well as to enable future work, we have created a reference experimental setup and trained and evaluated two baseline models.

A. Reference Experimental Setup

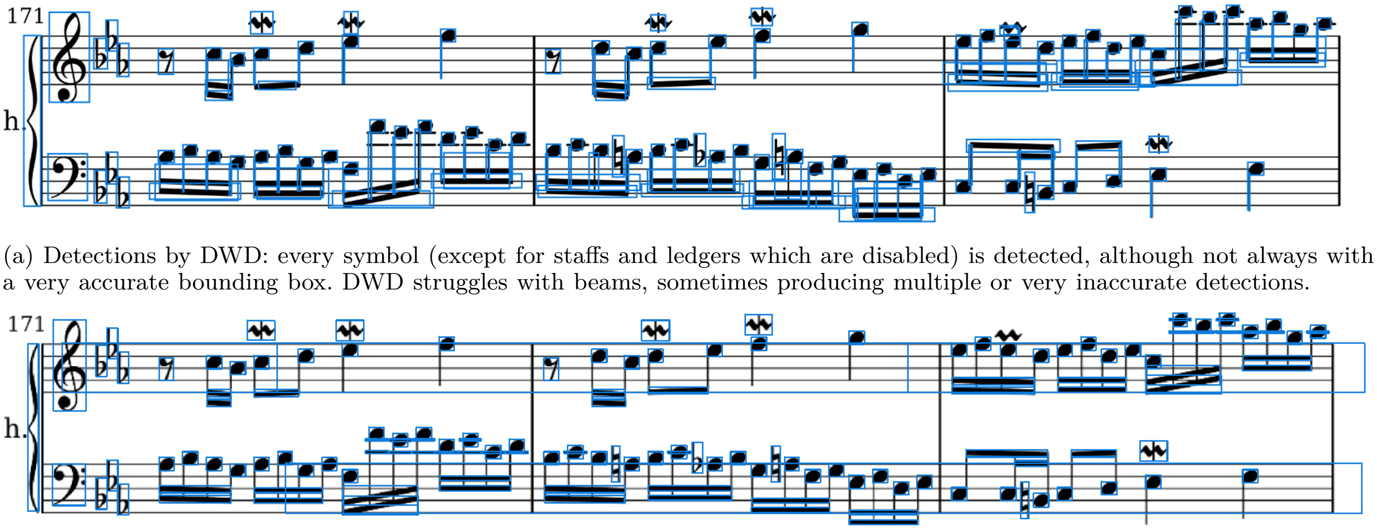
The results presented in this section are obtained by training the models using the train split of DeepScoresV2 dense until the training loss saturates. Previous experiments showed that due to training on random crops and the huge number of symbols, overfitting is not an issue. Both models are trained on the aligned (non-oriented) bounding boxes because there is currently no established method for oriented object detection in the OMR field. The results are reported using the metrics Average Precision at an overlap of 0.5 (AP0.5) [25] and COCO mean Average Precision (mAP) [26], computed by the evaluation function of the OBBAnns toolkit. Detailed information on the hyperparameters of the individual models are contained in the configuration files of the respective codebases.

B. Deep Watershed Detector

As a first baseline, we provide the Deep Watershed Detector (DWD) [2] that represents the current state of the art on DeepScores. We train it without any major modification from its originally published design: the only change is that we use the data at full resolution instead of applying a scaling factor of 0.5. Due to the large image size, this requires training on crops and also to perform inference using a crop and reassemble process that involves multiple forward passes. However, this can be done in a straight forward fashion since DWD is built entirely on fully convolutional neural networks [27]. For this baseline, we disable staff and ledger line detection because the DWD is by design unable to detect objects that share the same object centers.

C. Faster R-CNN

For the second baseline, we chose the Faster R-CNN architecture [28], based on the model published in Pacha et al. [6] that features specifically designed anchor boxes. As a backbone to the model, we use the newly introduced HRNet [29], which is able to produce extremely high-resolution features. This combination in itself is novel to the field of OMR.



(a) Detections by DWD: every symbol (except for staffs and ledgers which are disabled) is detected, although not always with a very accurate bounding box. DWD struggles with beams, sometimes producing multiple or very inaccurate detections.

(b) Detections by Faster R-CNN: all the stems are missed while other symbols are quite accurately detected.

Fig. 5: Example detection results of the two provided baseline models from the DeepScoresV2 dataset. Both are excerpts from full page detections, cropped for readability.

D. Evaluation and Discussion

Table IV presents class-wise average precision (AP) for both baselines. The combination of HRNet and Faster R-CNN appears to be have significant potential, achieving very high AP for almost all of the more common classes, and representing a new state of the art for music object detection on typeset music. The difference between mAP and AP0.5 is relatively low, which leads to the conjecture that the bounding box predictions are very accurate in terms of position and size. This can be visually confirmed by observing some Faster R-CNN detections as shown in Figure 5(b). Notably missing from the detections are stems, despite being the most common class of symbols. Further analysis is needed as to why these symbols are not properly detected.

The DWD consistently achieves lower average precision than Faster R-CNN except for the rarest symbols. It also has a bigger spread between mAP and AP0.5. A visual inspection of its detections in Figure 5(a) shows that it also finds all of the symbols but often with loose-fitting bounding boxes. Notably, the DWD detects the stems with a bounding box quality that is very usable in a practical setting, but too loose-fitting to impact the academic metric of AP0.5, let alone mAP. On the other hand, it is clear that DWD struggles considerably with the detection of beams, especially when they are at an angle.

These results show that both systems have their strengths and weaknesses. Currently, none is superior, although Faster R-CNN has made a big leap in performance thanks to the use of HRNet. The problems occurring with the beams further enforce the need for oriented bounding box annotations.

V. Conclusion and Future Work

We presented DeepScoresV2, an enhanced version of the DeepScores dataset for music object detection. DeepScoresV2 has a wider range of annotated symbols as well as oriented bounding boxes for more accurate and semantically informative detections. The presented baselines show that current models already perform quite well on DeepScoresV2, achieving a new state of the art, especially with a Faster R-CNN detector using an HRNet backbone. However, additional work is needed regarding small objects such as stems as well as rare objects. The newly provided ground truth for oriented bounding boxes can serve to develop new models that increase prediction accuracy on rotated objects with a non-uniform aspect ratio.

Evaluation metrics are designed with the goal of generating an accurate description of the performance of a model by a few representative numbers. There is a huge disparity between the metrics for the stem detections of DWD and how we judge the same detections of the stems visually (seen in figure 5a). This leads to the insight that AP0.5 and mAP, which have been designed for general object detection and only consider detections with an overlap of at least 50% between predicted and ground truth bounding boxes, do not fulfil this goal in every case, especially not for very small objects. We therefore conclude that AP0.5 and mAP are not well-suited to judge a music object detection systems and the field should strive to find or develop a more appropriate metric.

As DeepScoresV2 is a synthetic dataset, the images contained within are clean and have no noise. Therefore, models trained on DeepScoresV2 perform best on very clean scans. Building models that generalize well to scans

		DWD		Faster RCNN				DWD		Faster RCNN	
Class	No. Occ	mAP	AP0.5	mAP	AP0.5	Class	No. Occ	mAP	AP0.5	mAP	AP0.5
stem	65,088	0.000	0.003	0.004	0.013	keyboardPedalUp	144	0.049	0.180	0.490	0.571
noteheadBlackOnLine	34,785	0.502	0.880	0.934	0.973	rest32nd	140	0.483	0.965	0.992	0.993
noteheadBlackInSpace	33,923	0.489	0.872	0.933	0.969	fingering0	140	0.150	0.646	0.837	0.957
legerLine	23,809	0.000	0.000	0.656	0.854	fingering2	138	0.264	0.723	0.866	0.962
beam	18,846	0.030	0.114	0.819	0.919	fingering4	131	0.300	0.797	0.857	0.962
augmentationDot	5,525	0.035	0.151	0.765	0.871	dynamicS	127	0.043	0.212	0.813	0.945
staff	3,864	0.000	0.000	0.222	0.578	timeSig2	126	0.262	0.772	0.899	0.989
keySharp	3,478	0.448	0.942	0.882	0.967	timeSig1	116	0.349	0.830	0.906	0.997
keyFlat	3,188	0.443	0.921	0.881	0.946	clefCTenor	104	0.523	0.850	0.921	0.959
noteheadHalfOnLine	2,877	0.541	0.930	0.890	0.944	restWhole	94	0.175	0.698	0.069	0.085
noteheadHalfInSpace	2,810	0.510	0.907	0.852	0.913	keyboardPedalPed	93	0.029	0.097	0.563	0.706
tie	2,532	0.007	0.046	0.698	0.859	rest64th	93	0.286	0.669	0.983	0.989
rest8th	2,491	0.441	0.940	0.931	0.988	articStaccatissimoBelow	89	0.034	0.139	0.503	0.949
slur	2,430	0.042	0.159	0.771	0.881	rest128th	88	0.140	0.355	0.952	0.978
flag8thDown	2,281	0.442	0.895	0.926	0.986	articMarcatoAbove	88	0.127	0.545	0.390	0.509
clefG	2,203	0.430	0.880	0.927	0.992	fermataBelow	83	0.322	0.707	0.748	0.945
accidentalSharp	2,133	0.461	0.901	0.940	0.992	timeSig0	83	0.072	0.423	0.862	0.936
restQuarter	2,097	0.382	0.832	0.852	0.976	articTenutoAbove	82	0.007	0.050	0.410	0.685
accidentalNatural	1,941	0.318	0.826	0.900	0.984	ornamentMordent	81	0.286	0.762	0.931	0.988
flag8thUp	1,941	0.301	0.681	0.912	0.989	accidentalDoubleSharp	80	0.181	0.724	0.874	0.963
clefF	1,488	0.470	0.910	0.945	0.982	stringsUpBow	79	0.334	0.676	0.924	1.000
dynamicF	1,437	0.295	0.750	0.803	0.885	restDoubleWhole	77	0.136	0.509	0.896	0.972
timeSig4	1,349	0.361	0.696	0.653	0.723	ornamentTurn	71	0.110	0.577	0.961	1.000
articStaccatoAbove	1,193	0.061	0.250	0.745	0.891	arpeggiato	71	0.001	0.007	0.486	0.741
accidentalFlat	1,164	0.427	0.804	0.899	0.980	articMarcatoBelow	70	0.144	0.655	0.618	0.762
dynamicP	1,096	0.425	0.805	0.786	0.860	dynamicZ	70	0.332	0.974	0.906	0.991
noteheadWholeInSpace	1,008	0.306	0.808	0.868	0.911	timeSig9	69	0.100	0.459	0.908	1.000
repeatDot	876	0.017	0.067	0.833	0.989	stringsDownBow	66	0.548	0.962	0.966	1.000
noteheadWholeOnLine	865	0.387	0.919	0.890	0.939	clef15	63	0.015	0.088	0.627	0.839
rest16th	743	0.544	0.897	0.941	0.988	articStaccatissimoAbove	59	0.049	0.322	0.493	0.955
brace	725	0.000	0.000	0.869	0.969	noteheadDoubleWholeOnLine	57	0.052	0.351	0.372	0.650
restHalf	677	0.149	0.786	0.837	0.955	segno	55	0.471	0.945	0.969	1.000
dynamicM	533	0.292	0.782	0.698	0.807	ornamentTrill	52	0.420	0.943	0.856	0.997
articAccentAbove	521	0.369	0.871	0.818	0.960	flag32ndUp	49	0.231	0.674	0.502	0.810
articStaccatoBelow	503	0.017	0.078	0.641	0.790	coda	49	0.146	0.288	0.963	0.980
timeSig3	401	0.124	0.440	0.419	0.470	flag128thUp	45	0.035	0.185	0.947	0.999
flag16thDown	335	0.222	0.551	0.910	0.970	flag128thDown	42	0.030	0.288	0.948	1.000
tuplet3	329	0.092	0.362	0.765	0.941	flag64thDown	42	0.216	0.621	0.887	0.923
timeSig8	322	0.257	0.657	0.682	0.852	timeSig7	40	0.222	0.801	0.885	0.995
dynamicCrescendoHairpin	298	0.116	0.237	0.807	0.953	flag64thUp	29	0.028	0.095	0.802	0.850
articAccentBelow	274	0.398	0.864	0.776	0.963	articTenutoBelow	27	0.000	0.000	0.000	0.000
flag16thUp	263	0.370	0.813	0.937	1.000	restHBar	27	0.040	0.213	0.000	0.000
clefCAlto	255	0.396	0.649	0.903	0.970	ottavaBracket	26	0.000	0.000	0.173	0.300
flag32ndDown	239	0.010	0.017	0.000	0.000	tupletBracket	25	0.000	0.000	0.468	0.684
clef8	230	0.156	0.485	0.584	0.691	noteheadDoubleWholeInSpace	21	0.040	0.194	0.000	0.000
fingering1	226	0.081	0.307	0.860	0.959	ornamentTurnInverted	17	0.321	0.795	0.961	0.994
tuplet6	207	0.053	0.295	0.893	0.977	tuplet5	4	0.055	0.250	0.000	0.000
dynamicDiminuendoHairpin	192	0.053	0.153	0.747	0.918	dynamicR	4	0.088	0.125	0.000	0.000
timeSig6	185	0.197	0.794	0.461	0.574	fingering5	3	0.115	0.136	0.783	0.917
fermataAbove	184	0.227	0.741	0.846	0.966	tuplet1	0	0.000	0.000	0.000	0.000
keyNatural	183	0.265	0.721	0.867	0.993	tuplet8	0	0.000	0.000	0.000	0.000
timeSig5	146	0.007	0.044	0.009	0.014	accidentalDoubleFlat	0	0.000	0.000	0.000	0.000
caesura	146	0.041	0.204	0.757	0.916		mean	**0.203**	**0.503**	**0.700**	**0.799**
fingering3	146	0.137	0.443	0.852	0.947		weighted mean	**0.219**	**0.422**	**0.608**	**0.676**

TABLE IV: Classwise Average Precision at 0.5 overlap (AP0.5) as well as Mean Average Pecision (mAP) of DWD and Faster R-CNN.

of lower quality remain an important and open challenge. Our initial experiments have shown that simply printing and scanning known pages does not introduce enough meaningful real-world noise into the data to significantly impact generalizability. A more effective, but very expensive approach, would be to hand-label existing real-world data. The development of custom training and model architectures that promote better generalizability is, in our opinion, the most promising way to address this challenge.

We also encourage OMR researchers to use the newly available staff and rhythm information to build more powerful models that can directly infer higher-order semantic information.

Acknowledgment

The authors are grateful for the support through Innosuisse grant No. 34301.1 IP-ICT "RealScore", European Research Council Advanced Grant 742870, and the continued fruitful collaboration with ScorePad AG.

References

[1] J. Calvo-Zaragoza, J. Hajič Jr., and A. Pacha, "Understanding optical music recognition," ACM Comput. Surv., 2020.

[2] L. Tuggener, I. Elezi, J. Schmidhuber, and T. Stadelmann, "Deep watershed detector for music object recognition," in 19th International Society for Music Information Retrieval Conference, Paris, France, 2018, pp. 271–278.

[3] T. Stadelmann, M. Amirian, I. Arabaci, M. Arnold, G. F. Duivesteijn, I. Elezi, M. Geiger, S. Lörwald, B. B. Meier, K. Rombach et al., "Deep learning in the wild," in IAPR Workshop on Artificial Neural Networks in Pattern Recognition. Springer, 2018, pp. 17–38.

[4] I. Elezi, L. Tuggener, M. Pelillo, and T. Stadelmann, "Deepscores and deep watershed detection: current state and open issues," in 1st International Workshop on Reading Music Systems, Paris, France, 2018, pp. 13–14.

[5] J. Schmidhuber, "Deep learning in neural networks: An overview," Neural networks, vol. 61, pp. 85–117, 2015.

[6] A. Pacha, J. Hajič jr., and J. Calvo-Zaragoza, "A baseline for general music object detection with deep learning," Applied Sciences, vol. 8, no. 9, pp. 1488–1508, 2018.

[7] J. Calvo-Zaragoza, A. H. Toselli, and E. Vidal, "Handwritten music recognition for mensural notation with convolutional recurrent neural networks," Pattern Recognition Letters, vol. 128, pp. 115–121, 2019.

[8] Z. Huang, X. Jia, and Y. Guo, "State-of-the-art model for music object recognition with deep learning," Applied Sciences, vol. 9, no. 13, pp. 2645–2665, 2019.

[9] L. Tuggener, I. Elezi, J. Schmidhuber, M. Pelillo, and T. Stadelmann, "Deepscores - a dataset for segmentation, detection and classification of tiny objects," in 24th International Conference on Pattern Recognition, Beijing, China, 2018.

[10] A. Pacha, "The OMR datasets project," https://apacha.github.io/OMR-Datasets, 2017.

[11] J. Hajič jr. and P. Pecina, "The MUSCIMA++ dataset for handwritten optical music recognition," in 14th International Conference on Document Analysis and Recognition, Kyoto, Japan, 2017, pp. 39–46.

[12] J. Hajič jr., "Optical recognition of handwritten music notation," Ph.D. dissertation, Charles University, Prague, 2019.

[13] A. Pacha and J. Hajič jr., "The music notation graph (mung) repository," https://github.com/OMR-Research/mung, 2020.

[14] A. Fornés, A. Dutta, A. Gordo, and J. Lladós, "CVC-MUSCIMA: A ground-truth of handwritten music score images for writer identification and staff removal," International Journal on Document Analysis and Recognition, vol. 15, no. 3, pp. 243–251, 2012.

[15] J. Calvo-Zaragoza and J. Oncina, "Recognition of pen-based music notation: The HOMUS dataset," in 22nd International Conference on Pattern Recognition. Institute of Electrical & Electronics Engineers (IEEE), 2014, pp. 3038–3043.

[16] J. Calvo-Zaragoza and D. Rizo, "Camera-primus: Neural end-to-end optical music recognition on realistic monophonic scores," in 19th International Society for Music Information Retrieval Conference, Paris, France, 2018, pp. 248–255.

[17] M. Dorfer, J. Hajič jr., A. Arzt, H. Frostel, and G. Widmer, "Learning audio–sheet music correspondences for cross-modal retrieval and piece identification," Transactions of the International Society for Music Information Retrieval, vol. 1, no. 1, pp. 22–33, 2018.

[18] G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang, "Dota: A large-scale dataset for object detection in aerial images," in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.

[19] M. Everingham, S. M. Eslami, L. Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," Int. J. Comput. Vision, vol. 111, no. 1, pp. 98–136, Jan. 2015.

[20] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in Computer Vision – ECCV 2014. Cham: Springer International Publishing, 2014, pp. 740–755.

[21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in CVPR09, 2009.

[22] G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang, "Dota: A large-scale dataset for object detection in aerial images," in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.

[23] j. Jan Hajič and P. Pecina, "The MUSCIMA++ Dataset for Handwritten Optical Music Recognition," in 14th International Conference on Document Analysis and Recognition, ICDAR 2017, Kyoto, Japan, November 13 - 15, 2017, Dept. of Computer Science and Intelligent Systems, Graduate School of Engineering, Osaka Prefecture University. New York, USA: IEEE Computer Society, 2017, pp. 39–46.

[24] D. Spreadbury and R. Piéchaud, "Standard music font layout (SMuFL)," in First International Conference on Technologies for Music Notation and Representation - TENOR2015. Paris, France: Institut de Recherche en Musicologie, 2015, pp. 146–153.

[25] M. Everingham and J. Winn, "The pascal visual object classes challenge 2012 (voc2012) development kit," Pattern Analysis, Statistical Modelling and Computational Learning, Tech. Rep, vol. 8, 2011.

[26] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in European conference on computer vision. Springer, 2014, pp. 740–755.

[27] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3431–3440.

[28] R. Girshick, "Fast R-CNN," in Proceedings of the IEEE international conference on computer vision, 2015, pp. 1440–1448.

[29] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2019, pp. 5693–5703.